

## **DEINTERLEAVING APPARATUS AND METHOD FOR A DIGITAL COMMUNICATION SYSTEM**

### **CROSS-REFERENCE TO RELATED APPLICATIONS**

[01] This application claims the benefit of Korean Application No. 2003-16023, filed March 14, 2003, in the Korean Intellectual Property Office, the disclosure of which is incorporated herein by reference.

### **BACKGROUND**

#### **1. Field of the Invention**

[02] The present invention relates to a digital communication system, and more particularly to a deinterleaving apparatus and method for a digital communication system which improve reliability against burst errors occurring in transmission environments and simplify implementations thereof.

#### **2. Description of the Related Art**

[03] In general, multi-channel digital communication systems use the convolution code in order to eliminate random noise occurring upon transmissions. The convolution code is robust with respect to random noise, but weak on the burst error referring to errors clustered at a certain region. Accordingly, the multi-channel digital communication system converts data streams having burst errors into streams with random errors.

[04] The existing interleaving schemes are mainly classified into the block interleaving scheme and the convolutional interleaving scheme. The convolutional interleaving scheme has difficulties in its implementation compared to the block interleaving scheme, but has an advantage of being implemented in one fourth of the memory size and a half of the total delay time, so that the convolutional interleaving scheme is mainly used in instruments with a large interleaving size.

[05] The interleaver of the convolutional interleaving scheme (i.e., convolutional interleaver) is one that changes input data streams from a certain order to a random order by using an FIFO shift registers having different delay depths. The structures for the interleaver and deinterleaver of the convolutional interleaving scheme are shown in FIG. 1.

[06] The convolutional interleaver 110 is provided with  $k$  branches and  $m$ -bit FIFO shift registers  $M$ . That is, the larger the FIFO shift registers  $M$  become, the more the interleaving characteristics are enhanced. The convolutional deinterleaver 120 has the opposite structure to the interleaver 110. By doing so, interleaved data are deinterleaved so that original data is restored.

[07] The structure of a deinterleaver using such FIFO shift registers requires many logic gates, causing a problem of hardware size and complexity. As a technology developed to solve such a problem, the use of random access memories (RAMs) instead of the FIFO shift registers has been proposed.

[08] The use of the random access memories prevents the hardware from becoming large, but it requires additional logic elements to control address generations and memory controls.

[09] That is, in case of carrying out interleaving operations by use of the random access memories, a multi-channel digital broadcast system has a different delay depth for each channel, so that a receiver has to generate a different address according to the delay depth interleaved for each channel. In particular, enlarging interleaved delay depths for the improvement of digital broadcast quality brings out a problem of logic size and complicated implementation that generate an address for each channel.

[10] Further, depending upon deinterleaving operation characteristics, void data of different sizes is attached ahead of valid data for each channel, in accordance with a different interleaved delay depth for each channel. Accordingly, a starting point of valid data by channel becomes asynchronous with a system, causing a problem of additionally implementing synchronization logic elements.

[11] Such a conventional interleaver has a disadvantage that its size becomes large as its structure becomes complicated.

## SUMMARY

[12] In an effort to solve the problems described above, it is an aspect of an embodiment of the present invention to provide a deinterleaving apparatus and a method for a digital communication system enhancing reliability and simplifying implementations thereof.

[13] In order to achieve the above aspect and/or other features of the present invention, a deinterleaving apparatus for a digital communication system includes a row counter for increasing a row counting value based on input data, a column counter for increasing a column counting value every row period set in the row counter, plural synchronous counters corresponding to the row period, and for increasing a synchronous counting value every column period set in the column counter, an offset memory for storing offset values set in correspondence to interleaving delay depths of the input data by channel, and a deinterleaver memory for storing the input data at a write address generated based on the offset value. The input data stored in the deinterleaver memory is read at a read address generated based on the synchronous counting value.

[14] Preferably, the plural synchronous counters each have a different synchronous period from each other in correspondence to the row counting value, and a column period set in the column counter corresponds to a synchronous signal of the digital communication system.

[15] The input data includes valid and invalid data depending upon interleaving delay depths by channel, and the valid data is read at the read address generated based on the synchronous counting value.

[16] The read and write addresses are generated in combination of the row counting value, column counting value, and synchronous counting value, and the write address is generated based on the column counting value to which the offset value is added and the synchronous counting value.

[17] According to an embodiment of the present invention, a deinterleaving method for a digital communication system includes steps of increasing a row counting value of a row counter based on input data, increasing a column counting value of a column counter every row period set in the row counter, increasing synchronous counting values of plural synchronous counters in every column period set in the column counter, storing in a deinterleaver memory the input data at a write address generated based on offset values set in correspondence to interleaving delay depths of the input data by channel, and reading the input data stored in the deinterleaver memory at a read address generated based on the synchronous counting values.

[18] Plural row and column counters, which are necessary for the respective channels, are substituted by a single row counter and a single column counter, and as a result, constructions can be simplified. Further, by synchronizing the periods of row counter with the synchronous signal of a system, deinterleaving data, which are synchronized for each channel, can be obtained.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[19] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements, and wherein:

[20] FIG. 1 is a view for schematically showing structures of conventional interleaving and deinterleaving apparatuses;

[21] FIG. 2 is a view for schematically showing a structure of a deinterleaving apparatus for a digital communication system according to an embodiment of the present invention;

[22] FIG. 3 is a view for conceptually showing a structure of the deinterleaving apparatus of FIG. 2;

[23] FIG. 4A and FIG. 4B are views for explaining a process for input data to be written into a deinterleaver memory at write addresses generated by offset values set in the deinterleaving apparatus of FIG. 2;

[24] FIG. 5A and FIG. 5B are views for illustrating a process carrying out deinterleaving operations of a deinterleaving apparatus according to an embodiment of the present invention;

[25] FIG. 6A and FIG. 6B are timing diagrams for a synchronous signal and an output signal for a deinterleaving apparatus according to an embodiment of the present invention; and

[26] FIG. 7 is a flow chart for explaining a deinterleaving method for a digital communication system according to an embodiment of the present invention.

#### DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

[27] Hereinafter, the present invention will be described in detail with reference to 20 the accompanying drawings.

[28] FIG. 2 is a view for schematically showing a deinterleaving apparatus for a digital communication system according to an embodiment of the present invention. The deinterleaving apparatus has a row counter 210, a column

counter 230, plural synchronous counters 250, an offset memory 270, and a deinterleaver memory 290. Further, plural multiplexers are provided. The multiplexers each perform switching operations of selectively switching input signals for an output.

[29] The deinterleaver memory 290 has memory cells 291 each having  $c$  columns corresponding to a synchronous signal, and the memory cells 291 are arranged in  $k$  rows wherein  $k$  is at least one or more. Accordingly, an address system is configured with a combination of a row counting value, a column counting value, and a counting value, that is, synchronous counting value of the memory cells 291.

[30] The row counter 210 is a ring counter counting  $k$  rows, that is,  $R[0]$  to  $R[k-1]$ , and increases a row counting value in the order of input data.

[31] The column counter 230 is a ring counter counting  $c$  columns, that is, 0 to  $c-1$ , of the memory cell 291, and, if the row counter 210 completes counting up to  $k-1$ , increases a counting value of the column counter 230. The value of  $c$  is a value set in correspondence to a synchronous signal of a system.

[32]  $k$  synchronous counters 250 are provided, and the first through  $(k-1)$ th synchronous counters 251, 252, ... each count one or more memory cells 291 arranged in  $k$  rows  $R[0]$  to  $R[k-1]$ . That is, the synchronous counter 250 increases a counting value thereof if the column counter 230 completes counting up to  $c-1$  as a counting value thereof.

[33] For example, the first through (k-1)th synchronous counters 251, 252...each have a repetitive period as follows in accordance with deinterleaving operation characteristics.

[34] That is, the first synchronous counter 251 of row R[0] has a repetitive period of s, the second synchronous counter 252 of row R[1] has a repetitive period of s-1, and, in the same manner, the k-1 synchronous counter of row R[k-1] has a repetitive period of 1.

[35] The offset memory 270 stores offset values corresponding to interleaving delay depths by channel.

[36] In the offset values, when referring to the deinterleaver 120 of FIG. 1, an offset value of R[1] can be expressed in Formula 1 as follows:

[Formula 1]

$$offset = (k - 1)M - [(k - 1)M - (K - 2)M]$$

[37] That is, an offset value stored in advance is added to a column counting value and a synchronous counting value so that write addresses for input data by channel are generated.

[38] FIG. 3 is a view for conceptually explaining the deinterleaving apparatus of FIG. 2.

[39] The deinterleaving apparatus has the same row counter 210 and column counter 230 for multiple channels. The row counter 210 counts rows R[0] to R[k-1], and the column counter 230 counts columns [0] to [c-1] forming one memory cell 291. The k rows R[0] to R[k-1] have at least one or more memory cells 291 in correspondence to interleaving delay depths. For



example, R[0] has s(0 ~ s-1) memory cells, R[1] has s-1(0 ~ s-2) memory cells, ..., R[k-1] has 1(0) memory cells.

[40] The counters 210, 230, and 250 each have the operations as follows.

[41] That is, if the row counter 210 has a row counting value Row\_CNT reached to k-1, the column counter 230 increases a column counting value Column\_CNT by 1, and, if the column counter 230 has a column counting value Column\_CNT reached to c-1, the synchronous counter 250 increases a synchronous counting value Row\_CNT by 1. As such, write and read addresses for the deinterleaver memory 290 are generated in combination of the column counting value Row\_CNT, column counting value Column\_CNT, and synchronous counting value synch\_CNT counted by the respective counters 210, 230, and 250.

[42] First, a write address is generated by adding an offset value O by channel stored 10 in the offset memory 270 to the column counting value Column\_CNT and the synchronous counting value Synch\_CNT.

[43] Thereafter, the write address is generated in combination of the row counting values Row\_CNT and the column counting value Column\_CNT based on the synchronous counting value Synch\_CNT.

[44] FIG. 4A and FIG. 4B is a view for explaining a process through which the write addresses Row\_CNT, Column\_CNT, and Synch\_CNT are generated by offset values O<sub>1</sub> and O<sub>2</sub> by channel with input data pre-set.

[45] First, FIG. 4A is for a first channel having the biggest interleaving delay depth out of inputted multi-channel data.

[46] Valid data is inputted for the first input data of the first channel according to an interleaving result, and the first data is used for a write address (0,0,0) corresponding to an offset value  $O_{1-0}$  of the row  $R[0]$ .

[47] Invalid data is inputted for the second input data according to an interleaving result, and the second data is used for a write address (1, c-n, s-2) corresponding to an offset value  $O_{1-1}$  of the  $R[1]$ . That is, invalid data generated in an interleaving process is used for the write addresses from (1, c-n, s-2) to (1, c-1, s-2), and valid data is used from a write address (1, 0, 0).

[48] As above, valid data is first used for input data from a region with the synchronous counting value  $\text{Synch\_CNT}$  of '0' with respect to the respective rows  $R[0]$ ,  $R[1]$ , ...,  $R[k-1]$  based on the offset value  $O_1$  of the first channel stored in the offset memory 270 in advance.

[49] Accordingly, a read address for the first channel is generated by adding the row counting value  $\text{Row\_CNT}$  and the column counting value  $\text{column\_CNT}$  based on the synchronous counting value  $\text{Synch\_CNT}$  of '0', so that deinterleaving data of the first channel is outputted in synchronization.

[50] FIG. 4B is for a second channel having an interleaving delay depth smaller than that of the first channel.

[51] Valid data is inputted for the first input data of the second channel according to an interleaving result, and a write address (0,0,4) is used for the first data in correspondence to an offset value  $O_{2-0}$  of the row  $R[0]$ .

[52] Invalid data is inputted for the second input data according to an interleaving result, and a write address (1, c-m, 3) is used for the second data

in correspondence to an offset value  $O_{2,1}$  of the  $R[1]$ . That is, invalid data generated in an interleaving process is used for the write addresses from  $(1, c-m, 3)$  to  $(1, c-1, 3)$ , and valid data is used from a write address  $(1, 0, 4)$ .

[53] As aforementioned, valid data is first written for input data from a region with the synchronous counting value  $\text{Synch\_CNT}$  of '4' with respect to the respective rows  $R[0]$ ,  $R[1]$ , ...,  $R[k-1]$  based on the offset value  $O_2$  of the second channel stored in the offset memory 270 in advance.

[54] Accordingly, a read address for the second channel is generated in combination of the row counting value  $\text{Row\_CNT}$  and the column counting value  $\text{Column\_CNT}$  based on the synchronous counting value  $\text{Synch\_CNT}$  of '4', so the deinterleaving data of the second channel is outputted in synchronization.

[55] As described above, according to the present invention, input data of multiple channels which has different interleaving delay depths can be outputted as deinterleaving data in synchronization by channel.

[56] FIG. 5A, FIG. 5B, FIG. 6A, and FIG. 6B are exemplary views for explaining a deinterleaving process of a deinterleaving apparatus according to an embodiment of the present invention, and FIG. 7 is a flow chart for showing the deinterleaving process. Hereinafter, the deinterleaving process will be described in detail with reference to FIG. 5A to FIG. 7.

[57] FIG. 5A is a view for showing an interleaving apparatus 510 having five rows 20  $R[0]$ ,  $R[1]$ ,  $R[2]$ ,  $R[3]$ , and  $R[4]$  and one column  $C[0]$ .

[58] First, data inputted to the interleaving apparatus 510 is as follows:

[Table 1]

[Table 1]

Time	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	...
Data	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	...

[59] If the data of Table 1 is inputted to the interleaving apparatus 510, data interleaved in and outputted out of the interleaving apparatus 510 is as follows:

[Table 2]

Time	t1	t2	t3	t4	t5	t6	T7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	...
Data	D1	X1	X2	X3	X4	D6	D2	X5	X6	X7	D11	D7	D3	X8	X9	D16	D12	...

[60]  $X_1, X_2, X_3, X_4, \dots$  become invalid data generated in the interleaving process.

[61] An interleaver memory 520 has five rows R(0)(1)(2)(3)(4) and one column C(0) corresponding to the interleaving apparatus 510. Further, it is assumed that synchronous counting values have values from 0 to s-1 in case of R(0).

[62] FIG. 5B is a view for explaining a process through which interleaving data shown in Table 2 is written in the deinterleaver memory 520

[63] First, write addresses are generated by using offset values  $O_0, O_1, O_2, \dots$ , set in advance in correspondence to the interleaving apparatus 510.

[64] The offset value O, according to Formula 1 explained as above, is obtained by subtracting a corresponding delay depth R[0] from the biggest delay depth R[0], and then subtracting the difference of the above subtraction

from the biggest delay depth  $R[0]$ . Since the repetitive period of a column counter is 1, the  $M$  in Formula 1 becomes '1'.

[65] That is, offset values  $O$  stored in the offset memory 270 of the deinterleaving apparatus is as follows.

[66]  $(4 \times 1) - [(4 \times 1) - (4 \times 1)] = 4$  for the offset value  $O_0$  of row  $R(0)$ ,  $(4 \times 1) - [(4 \times 1) - (3 \times 1)] = 3$  for the offset value  $O_1$  of row  $R(1)$ ,  $(4 \times 1) - [(4 \times 1) - (2 \times 1)] = 2$  for the offset value  $O_2$  of row  $R(2)$ ,  $(4 \times 1) - [(4 \times 1) - (1 \times 1)] = 1$  for the offset value  $O_3$  of row  $R(3)$ , and  $(4 \times 1) - [(4 \times 1) - (0 \times 1)] = 0$  for the offset value  $O_4$  of row  $R(4)$ .

[67] As above, write addresses are generated based on the offset values  $O$  stored in advance in the offset memory 270, and a process through which the input data in Table 2 is written in the deinterleaver memory 520 is shown in FIG. 5B.

[68] Referring to Table 2, if a first input data  $D1$  is inputted, a write address  $(0, 0, 4)$  is generated by using a row counting value  $Row\_CNT$ , a column counting value  $Column\_CNT$ , a synchronous counting value  $Synch\_CNT$ , and an offset value  $O_0$  (S711). Accordingly, data  $D1$  is written in the write address  $(0, 0, 4)$  of the deinterleaver memory 520 (S713). Likewise, second data  $X1$  is used for a write address  $(1, 0, 3)$ , third data  $X2$  is used for a write address  $(2, 0, 2)$ , fourth data  $X3$  is used for a write address  $(3, 0, 1)$ , and fifth data  $X4$  is used for a write address  $(4, 0, 0)$ . Remaining interleaving data in Table 2 is used for corresponding addresses of the deinterleaver memory 520 in the same manner.

[69] Accordingly, data is sequentially written from a region having a synchronous counting value Synch\_CNT of '4' with respect to respective rows R(0), R(1), R(2), R(3), and R(4) of the deinterleaver memory 520.

[70] As above, with data written in the deinterleaver memory 520, read addresses are generated based on the synchronous counting value Synch\_CNT of '4' (S715). That is, generated are read addresses such as (0,0,4)(1,0,4)(2,0,4)(3,0,4)(4,0,4), (0,0,5)(1,0,5)(2,0,5)(3,0,5)(4,0,5), (0,0,6)(1,0,6)(2,0,6)(3,0,6)(4,0,6), ..., and, according to the generated read addresses, data (D1,D2,D3,D4,D5), (D6,D7,D8,D9,D10), ..., are read (S717).

[71] Accordingly, invalid data X1, X2, X3, X4, ..., generated upon interleaving is not generated, and input data is deinterleaved and outputted in order.

[72] FIG. 6A is a timing view of a system synchronous signal CLK, and FIG. 6B is a timing view for an output signal of data read from the deinterleaver memory 520 based on the synchronous counting value Synch\_CNT.

[73] As shown in FIG. 6A and FIG. 6B, data (D1, D2, D3, D4, D5) are read in synchronization with a system synchronous signal when Synch\_CNT = 4.

[74] In the manner as above, in synchronization with the synchronous counting value (Synch\_CNT = 4, 5, 6, ...), that is, a synchronous signal, data (D6, D7, D8, D9, D10), (D11, D12, D13, D14, D15),... are read from the deinterleaver memory 520 and then deinterleaved.

[75] According to the present invention, because the plural row and column counters, which are required for the respective channels, are substituted by a single row counter and a single column counter, the construction is simplified. Further, by synchronizing a period of the column counter with a system synchronous signal, deinterleaving data which are synchronized for the respective channels can be obtained. Therefore, any additional synchronization logic configuration for synchronization of deinterleaving data for the channels is not so necessary and as a result, the overall construction becomes simplified.

[76] According to the present invention, first, a deinterleaving apparatus can be simply implemented since the number of counters is minimized and any additional logic for synchronization is unnecessary.

[77] Second, by synchronizing a period of a column counter with a synchronous signal, deinterleaving data synchronized by channel can be obtained.

[78] Third, according to the minimum number of counters, the implementation of an apparatus for generating read and write addresses can also be simplified.

[79] While the invention has been shown and described with reference to a certain exemplary embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.